MATHEMATICS AND COMPUTATIONAL SCIENCES

Vol 4(1), 2023, pp:29-35
DOI:10.30511/mcs.2023.562662.1085
ISSN:2717-2708
http://mcs.qut.ac.ir/

Qom University of Technology

Facial recognition system using eigenfaces and PCA

Hamid Reza Yazdani [1], Ali Reza Shojaeifard [2]

Abstract: Face recognition is an essential field of image processing and computer vision. In this paper, we have developed a facial recognition system that can detect and recognize the face of a person by comparing the characteristics, and features of the face to those of known faces. Our approach considers the face recognition problem as an intrinsically two-dimensional recognition problem rather than requiring recovery of three-dimensional geometry, considering that eigenvectors generally describe human faces in the face space. The system works by projecting face images onto a feature space that spans the significant variations among known face images that are called eigenvectors (or principal components of the face set). Our technique can learn and recognize new faces in an unsupervised stylethis approach is based on eigenfaces and principal component analysis (PCA).

Keywords: Eigenfaces; Face recognition; Principal Component Analysis (PCA).
2020 Mathematics Subject Classification: 65F99; 15A69; 58C05.
Receive: 26 September 2022, **Accepted:** 15 March 2023

1 Introduction

The human face is the main focus of human society in social relations and plays a significant role in expressing identity, feelings, and emotions. Therefore, the ability to identify and estimate based on the face by artificial intelligence is crucial. We can see many faces throughout our lives, and recognize familiar faces between them at a later specified time. Computational models of face recognition are, in practice, very attractive, because they are both theoretically fascinating and helpful in many applications such as identifying criminals, security systems, image and video processing, and human-computer interactions. But these models are very difficult to develop because the faces are complex, multidimensional and visually have different meanings. The human face is a natural class of objects that have specific human meanings that may not be understood by the machine. These issues have made face recognition an area of high risk, which severely limits the ability of artificial intelligence systems to perform biological activities [5].

In a computer (as a machine), Images are expressed as a matrix that includes pixels, and each pixel has a particular color-coded in some numerical values. It is natural to ask if the computer can read the picture and understand it, and if so, whether we can describe the logic using matrix mathematics. To be less ambitious, people try to limit the scope of this problem to identifying human faces. An early attempt for face recognition is to suppose the matrix as high-dimensional details, and we infer a lower dimension

[1]Corresponding author: Department of Mathematics and Statistics, Imam Hossein Comprehensive University,Tehran, Iran, Email: hamidreza.yazdani@gmail.com
[2]Department of Mathematics and Statistics, Imam Hossein Comprehensive University, Tehran, Iran, Email: ashojaeifard@ihu.ac.ir

information vector from it, then try to recognize the person in the lower dimension. However, by exploring how to compress images to a much smaller size, we developed a skill to compare if two images portray the same human face even if the pictures are not identical [2].

In 1987, a paper by Sirovich and Kirby considered the idea that all images of the human face should be a weighted sum of a few *key pictures*, which is named the *eigen images*, as they are the eigenvectors of the covariance matrix of the mean-subtracted pictures of human faces. In the paper, they provided the algorithm of principal component analysis of the face image dataset in its matrix form, and the weights used in the weighted sum correspond to the projection of the face image into each eigen image [10].

In 1991, a paper by Turk and Pentland coined the term *eigenface*. They built this concept based on the idea of Sirovich and Kirby and used their weights, and eigen images as characteristic features to recognize faces. The paper by Turk and Pentland spread a memory-efficient way to compute the eigen images. It also proposed an algorithm about how the face recognition system can perform and contains updating the system to include new faces and combining it with a video capture system. A similar paper also suggested that the concept of eigenface can help reconstruct the partially obstructed images [10].

Recent advances in machine learning have made face recognition not a complicated problem. But in the past, researchers have made several efforts and developed some skills to make computers capable of identifying people. One of the preliminary attempts with medium success is eigenface, which is established on linear algebra techniques [3].

The rest of the paper is organized as follows: In section 2, some preliminaries and basic concepts from linear algebra, dimensionality reduction method, and eigenfaces are proposed, the proposed algorithm is introduced in section 3, and some implementations and results are reported in section 4, and finally, conclusions are considered in section 5.

2 Preliminaries

We have stated some preliminaries, concepts, and basics in this section.

Definition 2.1. *Assume that A is a squared matrix from rank n. The number λ is an eigenvalue of matrix A, if and only if $A - \lambda I$ is singular:*

$$det(A - \lambda I) = 0,$$

Where matrix I is a squared matrix from rank n. This equation is called characteristic equation and involves only λ. When A is n by n, characteristic equation has degree n. Then A has n eigenvalues, and by solving $(A - \lambda I)x = 0$, or $Ax = \lambda x$, for each λ, finding an eigenvector x [4], [6].

2.1 The Principal Component Analysis

As a statistical scheme, Principal component analysis (PCA) is a multivariate method that is designed for analyzing a data table in which several inter-correlated quantitative dependent variables describe observations. PCA employed an orthogonal transformation to convert a set of observations on probably correlated variables to a group of values for linearly uncorrelated variables. The target of this technique is to extract meaningful information and patterns from the table, to illustrate it as a set of new orthogonal variables called principal components, and to demonstrate the similar pattern of the observations and the variables as points in maps. The optimum linear least-squares decomposition is provided by this algorithm for a training set [8].

To estimate PCA presentation, every image is denoted by a point in $\mathbb{R}^{n \times m}$, where every image has n by m pixels. The optimum linear least-squares demonstration in $(N-1)$-D space, as well as the demonstration variance, are found using this algorithm [7]. The resultant demonstration is separated via a set of $N-1$ eigenvectors and eigenvalues. These eigenvectors orthonormal bases are *principal components* [8].

Algorithm 1. The subsequent steps should be followed in the PCA algorithm [8], [7]:

1) First, Calculate the covariance matrix S using the average value $E[x] = 0$. In this situation, the coexistence of covariance and autocorrelation matrices is observed, $R = E[xx^T] = S$; else, the average value is subtracted. Recall that, the autocorrelation matrix scheme for definite N-feature vectors, $x_i \in \mathbb{R}^l$, $i = 1, 2, ..., N$, can be shown as follows:

$$R = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T \tag{2.1}$$

2) Accomplish the S eigendecomposition and compute the l eigenvalues and associated eigenvectors λ_i, $a_i \in \mathbb{R}^l$, $i = 0, 2, ..., l-1$.
3) Sort the eigenvalues in descending order, i.e. $\lambda_0 \geq \lambda_1 \geq ... \geq \lambda_{l-1}$.
4) Choose the m greatest eigenvalues. Typically m is selected such that there is a large gap between λ_{m-1} and λ_m. Then, eigenvalues $\lambda_0, ..., \lambda_{l-1}$ are the m *principal components*.
5) Create the subsequent transformation matrix via the corresponding (column) eigenvectors a_i for $i = 0, 1, 2, ..., m-1$,
$$A = [a_0 \quad a_1 \quad a_2 \quad \cdots \quad a_{m-1}] \tag{2.2}$$
6) Commute every l-dimensional vector x of the primary space with an m-dimensional vector y using the transformation $y = A^T x$. Consequently, the ith element's variance, $E[y^2(i)]$ for $i = 0, 2, ..., l-1$, resembles to λ_i. Namely, the maximum variance can be maintained by choosing the elements that corresponded to the m greatest eigenvalues.

A scheme of the PCA technique applied to a data set of genes is shown in Figure 1.

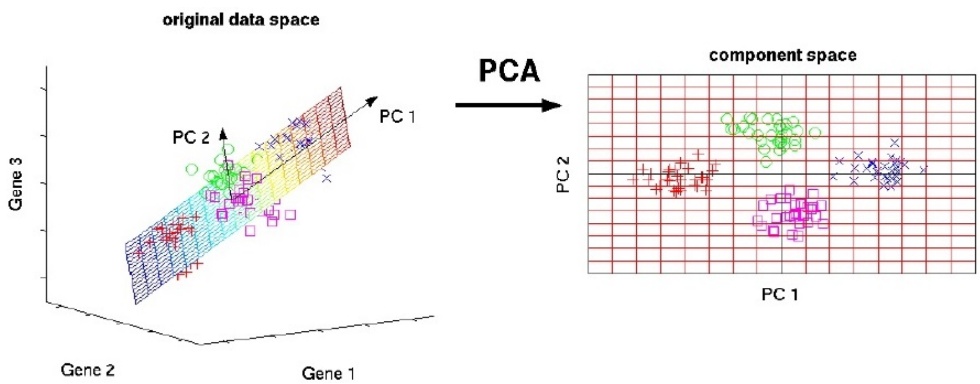

Figure 1: The scheme of the PCA method.

2.2 The Eigenface

Assume that we have numerous images of human faces in the same pixel dimension (e.g., all of them are $r \times c$ grayscale images). If we get M different images and vectorize each image into $L = r \times c$ pixels, we can illustrate the entire dataset as an $L \times M$ matrix (that is called matrix A), where each element in A is the grayscale value of pixel [10].

Note that PCA can be applied to any matrix, and resulting in some vectors (i.e., the principal components). The length of each principal component is the same as the column length of the matrix. The different principal components from the same matrix are orthogonal to each other (i.e., the vector dot-product of any two of them is zero). Consequently, the various principal components produced a vector space for which each column in the matrix can be represented as a linear combination (i.e., weighted sum) of the principal components [1].

Suppose that C is the matrix that subtracts each column of A with the mean vector a of A, i.e. $C = A - a$. Then a covariance matrix S as follows

$$S = C \cdot C^T,$$

Now, we find eigenvectors and eigenvalues of S. The principal components are these eigenvectors in the decreasing order of the eigenvalues. Because the matrix S is an $L \times L$ matrix, we may regard finding the eigenvectors of an $M \times M$ matrix $C \cdot C^T$ instead as the eigenvector v for $C \cdot C^T$ can be transformed into eigenvector u of $C \cdot C^T$ by $u = C \cdot c$, except we usually prefer to write u as normalized vector (i.e., the norm of u is 1) [5].

The physical meaning of the principal component vectors of A, or equivalently the eigenvectors of $S = C \cdot C^T$, is that they are the key directions in which we can construct the columns of a matrix A. We can keep only the first K principal component vectors. If matrix A is the dataset for face images, the first K principal component vectors are the top K most important *face images*. These K most important face images are called the *eigenface images* [10].

For any given face image, by utilizing the dot-product of vectors, we can project its mean-subtracted version onto the eigenface image. The proximity of the face image to the eigenface is the result. If the face image is unrelated to the eigenface, we would expect its result to be zero. For the K eigenfaces, we can find the K dot-product for any given face image. We can present the result as weights of this face image regarding the eigenfaces. The weight is given as a vector. Conversely, if we have a weight vector, we can add up each eigenface subjected to the weight and reconstruct a new face.

Lets denote the eigenfaces as a matrix F, which is an $L \times K$ matrix, and the weight vector is a column vector. Then we can construct the image of a face as $z = F \cdot W$, which z results in a column vector of length L. Because we are only using the top K principal component vectors, we should expect the resulting face image to be misinterpreted but retain some facial characteristics [5].

Since the eigenface matrix is constant for the dataset, changing the weight vector means varying face photos. Hence, the images of the same person have similar weight vectors, whereas these images are not identical. As a result, we can make use of the distance between two weight vectors (such as the L^2-norm) as a metric of similarity of two photos.[8].

3 The proposed algorithm

Algorithm 2. The steps of the Eigenfaces algorithm are as follows:

1) Each photo in the training set is displayed as linear combinations of Eigen image weights that are the base for the entire training set.

2) We use weights to exhibit the image in a smaller sub-space (The dimensionality reduction is constructed based on the K-best specific value for the base display).

3) To identify the face, we estimate the weights of the test image and compare it with the weights of all images in the training data set to determine which image has the smallest error between the test weights and the training set.

4 Implementation and results

We implemented our codes, algorithms, and experiments on the Asus Laptop with configuration as shown in table 1.

Table 1: The configuration of the experiment system.

CPU	Ci7-2670QM (8 Cores)
Frequency	2.2-3.1 GHz
RAM	16 GB (DDR3)
GPU	Nvidia Geforce GT 540M (2GB)
O.S.	Windows-10 Pro (64bit)
Software	MATLAB R2021a (64bit)

First, all trained images are imported and read, by flattening, every $m \times n$ matrix converted to a $n \times m \times 1$ vector. After that, an average face vector is calculated and subtracted from every image vector. All subtracted vectors are integrated into a matrix $A^{n\times m\times i}$ (where i is the number of images). The covariance matrix of A is calculated by $C = A \times A^T$. Now, we calculate eigenvalues and eigenvectors of matrix C. The K-largest of these eigenvectors is selected, such that $K << i$. Figure 2 shows the flowchart of the proposed algorithm.

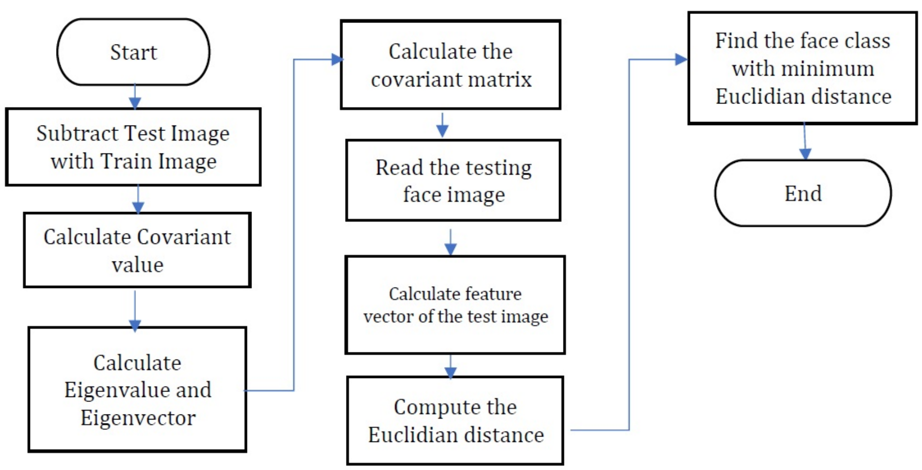

Figure 2: The flowchart of Eigenfaces.

Second, for testing the algorithm, the test image should be normalized by

$$I = R - S,$$

Where R is the test image vector, and S is the average face vector. After that, the weights are calculated and compared to existing weights in the eigenface datasets of the training image. Finally, the image is selected by the algorithm, which is the minimized differentiation between its weights and the weights of the tested image. Figure 3 shows the Eigenfaces algorithm in the environment of MATLAB software.

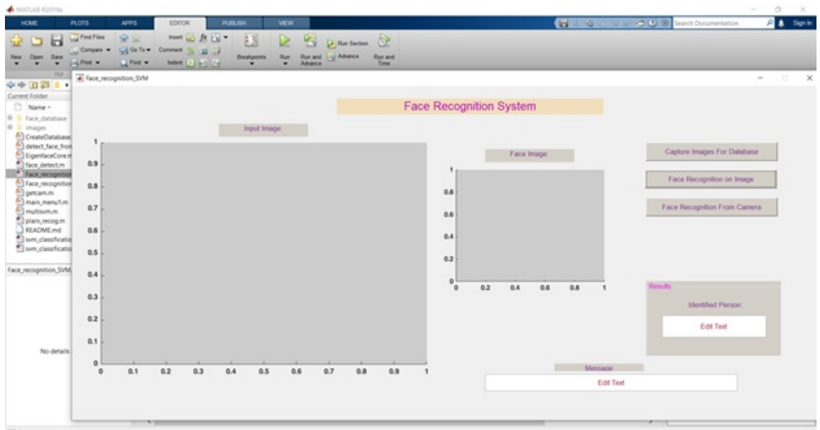

Figure 3: Face recognition system based on the Eigenfaces.

4.1 The error analysis

Studies show that algorithm error in face recognition increases with an increasing number of eigenvectors. Because the error in the pair of initial eigenvectors is small, the primary representation for the whole test set seems challenging, resulting in incorrect face recognition. In the number of eigenvectors above 10, the image is correctly identified, and the error becomes less and less. When the error decreases, the correct image is identified, and then convergence occurs. On the other hand, when we train the system with more photos, the convergence point will likely move to a bigger number of eigenvectors to represent the base of the face space. Error analysis is an excellent way to analyze the small face space to achieve accurate face recognition. Figure 4 shows the error analysis of the proposed algorithm implementation.

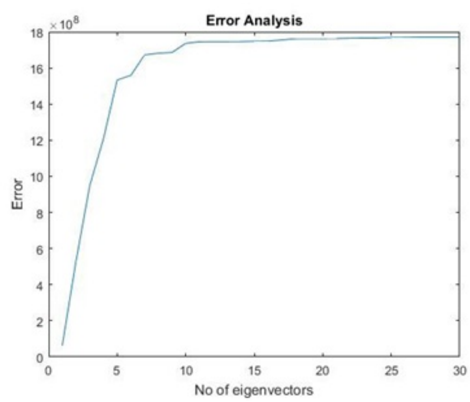

Figure 4: The error analysis of Eigenfaces algorithm.

5 Conclusion

In this paper, we introduce an algorithm based on eigenvectors and PCA, called Eigenfaces. This algorithm is stated and implemented for error analysis and performance comparison. The advantages of Eigenfaces are as follows:

1) This algorithm is easy to use and has a low computational cost.

2) The algorithm doesn't need any knowledge (such as facial features) about the image (except id).

and the limitations are as below:

1) A properly centered face is required for training/testing.

2) The algorithm is sensitive to exposure, shades, and the scale of the face in the image.

3) Eigenfaces need a front view of the face to work correctly.

Overall, this is an efficient way to identify a human face with suitable accuracy and speed, which has the strengths and weaknesses mentioned above.

References

[1] H. Abdi, L.J. Williams, Principal component analysis, WIREs Comp Stat, 2(4) 2010, 433-459.

[2] M. Agarwal, H. Agrawal, N. Jain, M. Kumar, Face Recognition Using Principle Component Analysis, Eigenface and Neural Network, International Conference on Signal Acquisition and Processing, 2 2010, 310-314.

[3] C.S. Chang, T.L. Liao, P.Y. Hsu, K.K. Chen, Human face recognition system using modified PCA algorithm and ARM platform, WIREs Comp Stat vol.2 (2010), Proceedings of Computer Communication Control and Automation (3CA), 294-297.

[4] C.D. Meyer, Matrix analysis and applied linear algebra, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2000.

[5] E.M Azriansyah, N. Hartuti, M. Fachrurrozi, B.A. Tama, A Study about Principle Component Analysis and Eigenface for Facial Extraction, J. Phys. Conf. Ser. 2 2019, 240-258.

[6] I. Gohberg, P. Lancaster, L. Rodman, Face Recognition Using Principle Component Analysis, Eigenface and Neural Network, Indefinite linear algebra and applications, Birkhuser Verlag, 2005, 310-314.

[7] P. Kamencay, R. Hudec, M. Benco, M. Zachariasova, 2D-3D Face Recognition Method Based on a Modified CCA-PCA Algorithm, International Journal of Advanced Robotics Systems, 2014, 215-230.

[8] H. Moon, P.J. Philips, Computational and performance aspects of PCA-based face recognition algorithms, Perception, 30 2001, 303-321.

[9] L. Sirovich, M. Kirby, Low-Dimensional Procedure for the Characterization of Human Faces, Journal of the Optical Society of America A, 4(3) 1987, 519-524.

[10] M.A. Turk, A.P. Pentland, Face recognition using eigenfaces, Proceedings of 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, 586-591.